UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/648,577	08/25/2003	Namit Jain	50277-2236	4090

29989 7590 07/14/2009
HICKMAN PALERMO TRUONG & BECKER, LLP
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110

EXAMINER	
RADTKE, MARK A	

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
07/14/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte NAMIT JAIN, ELLEN BATBOUTA, NIPUN AGARWAL, RAVI MURTHY, PAUL REILLY, and JAMES STENOISH

Appeal 2008-004465
Application 10/648,577
Technology Center 2100

Decided: July 14, 2009[1]

Before LANCE LEONARD BARRY, JAY P. LUCAS, and THU ANN DANG, *Administrative Patent Judges.*

DANG, *Administrative Patent Judge.*

DECISION ON APPEAL

Appellants appeal the Examiner's final rejection of claims 1-20 under 35 U.S.C. § 134. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm-in-part.

[1] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Data (electronic delivery).

I. STATEMENT OF THE CASE

A. INVENTION

According to Appellants, the present application discloses techniques and systems "for storing instances of an opaque type in a database according to a direct path loading approach" (Spec. 26, ll. 2-3).

B. ILLUSTRATIVE CLAIM

> 1. A method of storing data into a database, the method comprising:
>
> a loader application receiving data;
>
> determining one or more routines that are associated with a type of said data, wherein said one or more routines are implemented by a program that is external to both said loader application and a database server that manages said database;
>
> invoking said one or more routines;
>
> in response to said one or more routines being invoked, said program performing steps comprising:
>
> creating a data structure that has one or more elements that correspond to one or more attributes of said type; and
>
> populating said one or more elements with one or more values that are specified in said data, wherein said one or more values correspond to said one or more attributes;

generating, based on said data structure, a data stream that conforms to a format of data blocks of said database; and

writing said data into one or more data blocks in said database.

C. REJECTIONS

The Examiner relies upon the following prior art in rejecting the claims on appeal:

Skinner	US 6,085,198	July 4, 2000

O'Reilly, *What is the Direct Path?*, Oracle SQL*Loader: The Definitive Guide, Ch. 10, Sec. 10.1, (2001), http://proquest.safaribooksonline.com/1565929489/copyright (hereinafter "O'Reilly").

Claims 1-3, 5-11, and 13-20 stand rejected under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claims 4 and 12 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Skinner and O'Reilly.

II. ISSUES

1) Have Appellants shown the Examiner erred in finding Skinner discloses a "program" that is "external to both said loader application and a database server," implements "one or more routines that are associated with a type of said data," and, "in response to said one or more routines being invoked," performs the step of "creating a data structure that has one or

more elements that correspond to one or more attributes of said type" (claim 1)?

2) Have Appellants shown the Examiner erred in finding Skinner discloses the step of "generating, based on said data structure, a data stream that conforms to a format of data blocks of said database; and writing said data into one or more data blocks in said database" (claim 1)?

III. FINDINGS OF FACT

Skinner

1) Elements of Skinner's client, application, and database tiers may be executed within separate computers interconnected by a network (col. 2, ll. 29-36).

2) Skinner loads extracted metadata into an "import data structure such as a Vector [sic]," which Skinner describes as an array of elements that may be of any data type (col. 18, ll. 6-39).

3) Skinner's data store management component 308 obtains metadata, e.g., from a vector, to generate SQL "create table" commands (col. 16, ll. 38-44; col. 37, ll. 16-58; and col. 38, ll. 18-21).

4) An SQL "create table" command specifies the table name and attributes that make up a database table and is submitted to the database server 311 to configure the database (col. 38, ll. 18-21).

5) The database may be organized such that, for a database table, each column represents a particular data attribute and each row

represents a set of attribute values (col. 13, ll. 13-18).

IV. PRINCIPLES OF LAW

Claim Interpretation

"[T]he PTO gives claims their 'broadest reasonable interpretation.'" *In re Bigio*, 381 F.3d 1320, 1324 (Fed. Cir. 2004) (quoting *In re Hyatt*, 211 F.3d 1367, 1372 (Fed. Cir. 2000)). However, "limitations are not to be read into the claims from the specification." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citing *In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989)).

When descriptive material is not functionally related to the substrate, the descriptive material will not distinguish the invention from the prior art in terms of patentability. *In re Ngai*, 367 F.3d 1336, 1339 (Fed. Cir. 2004). *Cf. In re Gulack*, 703 F.2d 1381, 1385 (Fed. Cir. 1983).

35 U.S.C. § 102

Under 35 U.S.C. § 102, "[a] single prior art reference that discloses, either expressly or inherently, each limitation of a claim invalidates that claim by anticipation." *Perricone v. Medicis Pharm.Corp.,* 432 F.3d 1368, 1375 (Fed. Cir. 2005) (citation omitted).

35 U.S.C. § 103

Section 103 forbids issuance of a patent when "the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). In *KSR*, the Supreme Court emphasized "the need for caution in granting a patent based on the combination of elements found in the prior art," and discussed circumstances in which a patent might be determined to be obvious. *Id.* at 415 (citing *Graham v. John Deere Co.*, 383 U.S. 1, 12 (1966)). The Court reaffirmed principles based on its precedent that "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *Id.* at 416. The operative question in this "functional approach" is thus "whether the improvement is more than the predictable use of prior art elements according to their established functions." *Id.* at 417.

In affirming a determination of obviousness, the Federal Circuit has relied, in part, on an applicant's failure to present evidence that the modification was "uniquely challenging or difficult for one of ordinary skill in the art" or "represented an unobvious step over the prior art." *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418-19).

V. ANALYSIS

35 U.S.C. § 102

Claims 1-3, 5, 6, 9-11, 13, and 14

With respect to claim 1, Appellants argue the Examiner fails to cite a feature of Skinner that teaches all limitations of the recited "program." (App. Br. 5). More particularly, they argue:

> Claim 1 requires (a) that this "creating" step be performed by "said program" ("said program performing steps comprising: creating"), (b) that "said program" must be "a program that is external to both said loader application and a database server that manages said database," and (c) that the "routines," in response to whose invocation the "program" performs the "creating," must be "implemented by" that same "program." Even if Skinner's approach creates database tables in a database, Skinner does not disclose, teach, or suggest that these database tables are created by a program that is external to a database server or in response to the invocation of routines that are implemented by such an external program.

(App. Br. 5; Appellants' emphasis).

The Examiner does not cite a specific component of Skinner as teaching the "program" of claim 1. He appears to rely on "distinct" programs of Skinner's database server 311, stating, "If the database server were considered to be a software server application (such as an Oracle database), then the program of Skinner would be a distinct software program, and thus 'external' (logically) to the loader application and the database server" (Ex. Ans. 9).

In view of the above, an issue we address on appeal is whether Skinner discloses a "program" that is "external to both said loader application and a database server," implements "one or more routines that are associated with a type of said data," and, "in response to said one or more routines being invoked," performs the step of "creating a data structure that has one or more elements that correspond to one or more attributes of said type" (claim 1).

We begin our analysis by giving the claims their broadest reasonable interpretation. *See In re Bigio*, 381 F.3d at 1324. Furthermore, our analysis will not read limitations into the claims from the specification. *See In re Van Geuns*, 988 F.2d at 1184.

Claim 1 simply does not place any limitation on what the term "external" is to be, is to represent, or is to mean, other than that the "program" must be "external to both said loader application and a database server." Thus, we interpret the "external program" of claim 1 to be a program that is separate and distinct from the "loader application" and "database server."

Skinner discloses that its data store management component may be separated, by networks, from other elements of Skinner's client, application, and database tiers (FF 1). That is, the data store management component of Skinner is separate and distinct from other components of the client, application and database tiers. An artisan would have therefore understood the data store management component as being "external to both said loader application and a database server" (claim 1).

To create database tables, Skinner's data store management component obtains metadata, e.g., from vectors, and generates SQL "create table" commands from the metadata (FF 3). Because the data store management component obtains metadata, an artisan would have understood the component to implement "one or more routines that are associated with a type of said data." Further, because the data store management component generates the "create table" commands from the obtained metadata, an artisan would have understood the component to perform the "creating a data structure" step "in response to said one or more routines being invoked" (claim 1).

Thus, for the above reasons, we find that Skinner discloses a "program" that is "external to both said loader application and a database server," implements "one or more routines that are associated with a type of said data," and, "in response to said one or more routines being invoked," performs the step of "creating a data structure that has one or more elements that correspond to one or more attributes of said type" (claim 1).

In addition to arguing that Skinner does not disclose the "program" of claim 1, Appellants argue that "Skinner doesn't disclose that any 'data stream,' which 'conforms to a format of the data blocks of the database,' is 'generated based on' a database table (the alleged 'data structure') that has already been created in the database" (App. Br. 6).

However, the Examiner responds that the "generating" and "writing" steps "merely require writing data to a database" and "may be carried out by any part of the system" of Skinner (Ex. Ans. 10).

Accordingly, an issue we address on appeal is whether Skinner discloses the steps of "generating, based on said data structure, a data stream that conforms to a format of data blocks of said database; and writing said data into one or more data blocks in said database" (claim 1).

Giving claim 1 its broadest reasonable interpretation, we agree with the Examiner's statement that the "generating" and "writing" steps merely require writing data to a database and may be carried out by any part of the system. Contrary to Appellants' argument, which is not commensurate with the language of claim 1, these steps of claim 1 do not require the recited "data stream" to be generated based on a data structure "that has already been created in the database" (App. Br. 6).

Skinner discloses that an SQL "create table" command specifies the table name and attributes that make up a database table and is submitted, via the data store management component, to the database server to configure the database (FF 4). The database may be organized such that, for a database table, each column represents a particular data attribute and each row represents a set of attribute values (FF 5).

Because Skinner's system submits an SQL "create table" command to the database server and, in that submission, specifies attributes corresponding to blocks of a database table, an artisan would have understood the system as "generating, based on said data structure, a data stream that conforms to a format of data blocks of said database" (claim 1). Further, an artisan would have understood the system as "writing said data into one or more data blocks in said database" (claim 1).

Accordingly, we conclude that Appellants fail to show that the Examiner erred in rejecting claim 1 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Appellants do not provide separate arguments for claims 2, 3, 5, 6, 9-11, 13, and 14. Therefore, Appellants fail to show that the Examiner erred in rejecting claims 2, 3, 5, 6, 9-11, 13, and 14 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claims 7 and 15

Claim 7 depends from claim 1 and recites "adding, to a table, an entry that indicates an association between said type and said one or more routines." However, the phrase "that indicates an association between said type and said one or more routines" is nonfunctional descriptive material that is not functionally related to the claimed method of storing data. That is, the phrase "that indicates an association" does not change the functionality of or provide an additional function to the adding step of claim 1, but is merely a description of the entry of the table being added in the adding step. When descriptive material is not functionally related to the claimed embodiment, the descriptive material will not distinguish the invention from the prior art in terms of patentability. *See In re Ngai* at 1339 and *In re Gulack* at 1385. Thus, we interpret claim 7 as merely requiring the step of "adding, to a table, an entry."

The Examiner cites Skinner's "myMethods" vector as teaching the limitations of claim 7 (Ex. Ans. 5 and 11). Skinner describes a vector as being an array of elements that may be of any data type (FF 2). An artisan would have understood Skinner's use of the "myMethods" vector, an array of elements, as teaching the limitation of "adding, to a table, an entry" (claim 7).

Accordingly, Appellants fail to show that the Examiner erred in rejecting claim 7 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claim 15 recites "instructions which . . . causes [sic] the one or more processors to perform the method recited in Claim 7." Appellants do not provide separate arguments for claim 15. Therefore, Appellants fail to show that the Examiner erred in rejecting claim 15 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claims 8 and 16

Claim 8 depends from claim 1 and recites "said invoking comprises invoking one or more routines that are located at one or more addresses that are associated with said type via an associative structure." However, the phrase "that are located at one or more addresses that are associated with said type via an associative structure" is nonfunctional descriptive material that is not functionally related to the claimed method of storing data. That is, the phrase "that are located at" does not change the functionality of or provide an additional function to the "invoking" step but is merely a description of the "routines" being invoked in the invoking step. When the descriptive material is not functionally related to the claimed embodiment, the descriptive material will not distinguish the invention from the prior art in terms of patentability. *See In re Ngai* at 1339 and *In re Gulack* at 1385. Thus, we interpret claim 8 as merely requiring the act of "invoking one or more routines" in performing "said invoking" step of base claim 1.

The Examiner cites Skinner's loading of extracted metadata into data structures (Ex. Ans. 6; citing col. 18, ll. 6-10), such as vectors (Ex. Ans. 12), as teaching the limitations of claim 8. Skinner indeed stores extracted metadata into vectors (FF 2). Further, Skinner's data store management component obtains such metadata, e.g., from a vector, to generate the SQL "create table" commands (FF 3). An artisan would have therefore understood the use of such vectors, to obtain the metadata and generate the SQL "create table" commands, as teaching the limitation of "invoking one or more routines" (claim 8).

Thus, Appellants fail to show that Skinner does not disclose the limitations of claim 8. Accordingly, Appellants fail to show that the Examiner erred in rejecting claim 8 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claim 16 recites "instructions which . . . causes [sic] the one or more processors to perform the method recited in Claim 8." Appellants do not provide separate arguments for claim 16. Therefore, Appellants fail to show that the Examiner erred in rejecting claim 16 under 35 U.S.C. § 102(b) as anticipated by Skinner.

Claims 17 and 19

Claim 17 depends from claim 1 and recites "said program registering, with said loader application, said one or more routines, which are not implemented by said loader application." The Examiner cites Skinner's "reflection methods" as teaching this limitation of claim 17 (Ex. Ans. 6). Appellants argue, "Skinner does not disclose that a program registers, with a loader application, routines having such characteristics [as the "routines" of claim 1] (e.g., routines that create and populate a data structure as discussed above with reference to Claim 1).” (App. Br. 10-11).

Based upon the record on appeal, we agree with Appellants that the Examiner has not shown that Skinner’s “reflection methods” disclose the step of “said program registering, with said loader application, said one or more routines” (claim 17). For example, the relied-upon sections do not show that Skinner’s “reflection methods” cause a program to register routines with Skinner’s communication management component, which is cited as teaching the “loader application” of base claim 1 (Ex. Ans. 4).

As such, we reverse the rejection of claim 17. We thus conclude that Appellants have shown that the Examiner erred in rejecting claim 17, and dependent claim 19 falling therewith, under 35 U.S.C. § 102(b).

Claims 18 and 20

Claim 18 depends from claim 1 and recites “said loader application invoking at least one of said one or more routines to find out (a) a number of one or more attributes within an opaque type and (b) one or more types of said one or more attributes within said opaque type.” As with claim 17, the Examiner cites Skinner’s “reflection methods” as teaching this limitation of claim 18 (Ex. Ans. 7 (citing col. 19, ll. 14-17) and 13). Appellants argue, “Skinner does not disclose that the ‘reflection methods’ that the Examiner analogizes to the ‘said one or more routines’ in the rejection of Claim 18 have such characteristics [as the routines of claim 1] (e.g., routines that create and populate a data structure as discussed above with reference to Claim 1)" (App. Br. 11-12).

Based upon the record on appeal, we agree with Appellants that the Examiner has not shown that Skinner's "reflection methods" disclose the step of "said loader application invoking at least one of said one or more routines to find out (a) a number of one or more attributes within an opaque type and (b) one or more types of said one or more attributes within said opaque type" (claim 18). For example, the relied-upon sections do not show that Skinner's "reflection methods" cause the communication management component, which is cited as teaching the "loader application" of base claim 1 (Ex. Ans. 4), to invoke routines with the characteristics "(a)" and "(b)" of claim 18; or to invoke routines that create and populate a data structure (as required by claim 1). Thus, those sections do not teach the limitations of claim 18.

As such, we reverse the rejection of claim 18. We thus conclude that Appellants have shown that the Examiner erred in rejecting claim 18, and dependent claim 20 falling therewith under 35 U.S.C. § 102(b).

Claims 4 and 12

Claim 4 depends from claim 1 and recites "said creating, said populating, said generating, and said writing are performed using a direct path loading approach rather than a conventional path loading approach, and without causing a Structured Query Language (SQL) engine to load said data." With respect to claims 4 and 12, Appellants acknowledge "that direct path loading, in general, was known prior to the filing of the present application," but argue that "at the time of Skinner and O'Reilly, people of ordinary skill in the art did not know how to load data of opaque types, whose structure was not known to the loader, into a database using the direct path loading approach" (App. Br. 12-13). However, the Examiner finds:

> Direct path loading is a well known technique that would have yielded predictable results to one of ordinary skill in the art at the time of the invention. There are no facts that would lead one to expect that opaque types, properly serialized by Skinner, could not be processed for direct path loading.

(Ex. Ans. 13).

We agree with the Examiner's finding that "[t]here are no facts that would lead one to expect that opaque types … could not be processed for direct path loading" (Ex. Ans. 13; Board's emphasis). Appellants have presented no evidence that the inclusion of direct path loading in Skinner's system would have been "uniquely challenging or difficult for one of ordinary skill in the art," *Leapfrog*, 485 F.3d at 1162, or would yield "more than one would expect from such an arrangement," *KSR*, 550 U.S. at 417.

Accordingly, we affirm the rejection of claim 4 and claim 12 falling therewith under 35 U.S.C. § 103(a) as unpatentable over Skinner and O'Reilly.

CONCLUSIONS

(1) Appellants have not shown the Examiner erred in finding that claims 1-3, 5-11, and 13-16 are anticipated by Skinner.

(2) Appellants have shown the Examiner did not meet his burden in finding that claims 17-20 are anticipated by Skinner.

(3) Appellants have not shown the Examiner erred in finding that claims 4 and 12 are unpatentable over Skinner and O'Reilly.

(4) Claims 1-16 are not patentable.

DECISION

We affirm the Examiner's decision rejecting claims 1-3, 5-11, and 13-16 under 35 U.S.C. § 102(b).

We reverse the Examiner's decision rejecting claims 17-20 under 35 U.S.C. § 102(b).

We affirm the Examiner's decision rejecting claims 4 and 12 under 35 U.S.C. § 103(a).

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED IN PART

PEB

HICKMAN PALERMO TRUONG & BECKER, LLP
2055 GATEWAY PLACE
SUITE 550
SAN JOSE CA 95110